# 使用 git 创建工作文件夹并运行 pthread 代码

### 创建 git 工程

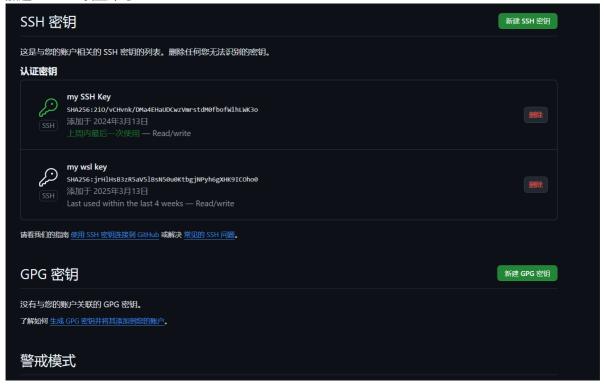
使用 sudo mkdir os-homework 命令在合适的位置创建工作文件夹

使用 git init 命令初始化 git 工程,此时默认主分支为 master

使用 git branch -m main 将当前主分支修改为 main, 以便于和远程仓库统一

使用 git remote add origin git@github.com:markzhang12345/os-homework.git 将远程仓库添加 进 git 工程

这里我使用的是 ssh 的方式进行连接,使用 git 自带的 ssh 工具可以使用邮箱生成本机 ssh,将公钥添加进 GitHub 设置即可



### 提交更改

使用 git add <文件>即可将已完成的更改提交至暂存区,一般直接使用 git add . 即可方便的将当前目录下的所有已修改的文件提交至暂存区

完成所有修改并提交至暂存区后,使用 git commit -m "<说明>" 即可将完成的修改提交至当前分支之后,使用 git push origin main 即可将当前分支的更新提交至远程仓库

#### 新建分支

如果正在进行关键重构,可以新建分支进行修改,避免在主分支发生不可逆的意外事故可以使用 git branch -b <分支名> 来新建分支,并使用 git checkout <分支名> 来切换当前所在的分支

在想要将其他分支合并至当前分支时,使用git merge <其他分支> 命令即可

## 编译并运行 pthread 代码

使用 cd ./os\_homework\_3/codes 命令进入工作文件夹,使用 sudo touch Pthreads\_Hello\_World.c 命令新建 .c 文件,使用 sudo vim Pthreads\_Hello\_World.c 在文本编辑器中编写代码如下

```
#include <pthread.h>
void* worker(void* arg) {
    printf("Hello World\n");
return NULL;
int main() {
    pthread_t tid;
    if (pthread_create(&tid, NULL, worker, NULL)) {
       printf("Cannot create thread\n");
        exit(1);
    printf("Main waits for thread to finish...\n");
    pthread_join(tid, NULL);
    exit(0);
"Pthreads_Hello_World.c" [noeol][dos] 21L, 402B
```

使用 sudo gcc -o Pthreads\_Hello\_world Pthreads\_Hello\_world.c 命令编译代码,并使用 ./Pthreads\_Hello\_world 运行代码,显示如下输出

kpmark@LAPTOP-AITJPLGC:/mnt/d/os-homework/os\_homework\_3/codes\$ ./Pthreads\_Hello\_World
Main waits for thread to finish...
Hello World